Chapter 19

System Architecture and Administration

The CM-5 system architecture provides for multiple task execution partitions, I/O devices, and fault detection and recovery. It supports a centralized system administration facility that gives the administrator flexibility to optimize the use of system resources. All these tasks are handled through various extended capabilities and privileged features of the Control Network and Data Network, with the assistance of a third network, the Diagnostic Network.

19.1 The System Console

Administration is managed from a System Console, a process executing on a control processor that has a Diagnostics Network interface. Large CM-5 systems will typically have a dedicated processor for administration; on small CM-5 systems, the administration process may run on a control processor that also has other tasks.

The System Console processor has a Diagnostics Network connection that allows it to address the entire system. It is responsible for configuring the system on power-up, for partitioning the system, and for managing the system as it changes due to repartitioning and hardware failures. A database containing the status of the overall system, kept up to date by the Diagnostics Network, helps it perform these tasks.

Copyright © 1993 Thinking Machines Corporation

19.2 Allocation of Resources

The CM-5 system provides flexible allocation of computational resources. The administrator can subset processing nodes into partitions; the administrator can also allocate control processors to single or multiple I/O devices.

Partitions

The set of computational and network resources in use at any given instant by a single user task is called a *partition*. Each partition constitutes a complete task execution system that may be used for timesharing, batch processing, or both.

The system administrator creates partitions dynamically, to best accommodate the site's workload. Some administrators may use a partitioning strategy that involves changing the partitioning two or three times during the course of a day. Other sites may stick with a single set of partitions for several days at a time.

An administrator might, for example, create three partitions on a system: one dedicated to a production run of a single large application, a second one used for timeshared program development by day and scheduled batch processing by night, and a third small one dedicated to around-the-clock timeshared access.

All partitions are joined by the Control Network and Data Network in a single integrated system. Resources can therefore be reallocated from one partition to another when necessary. For example, all partitions might be joined to form one giant partition in order to tackle a single giant application. As another example, if processors were to fail in the partition dedicated to a production run, they could be replaced (by reconfiguring the networks) with processors borrowed from another partition. The production run could then be rolled back to a prior checkpoint and resumed with minimal disruption, while the failed processors were powered down and, at a convenient time, physically replaced.

I/O

I/O devices and interfaces, like processing nodes, reside in specific areas of the network address space and are managed by control processors. The I/O resources they control are available to processes running on any partition. The Data Network transfers data between I/O devices and partitions, while the Control

November 1993
Copyright © 1993 Thinking Machines Corporation

Network is used by the operating system to monitor the transfers and signal errors.

19.3 Partitions and Networks

From a system view, the Control Network and Data Network are designed to provide

- the capability for flexible partitioning of computing resources
- the isolation of each partition's network activity
- high throughput for all cases of data transfer

To see how this works, we look at the way in which the address space on these two networks is managed.

Figure 34 shows a simplified view of address space management in the networks. As this figure suggests, each of the superficially homogeneous networks is logically split by hardware-supported, software-configured mechanisms so as to devote a portion to each partition or I/O resource. Additional network capacity is dedicated to carrying traffic between the various partitions and devices that make up the system at any given time. Network resources allotted to one partition do not overlap those associated with another. Moreover, traffic from one partition to another, or between a partition and an I/O device, consumes no network resources belonging to any intervening partition. The network design thus guarantees that network traffic within one partition cannot affect the behavior or the performance of traffic in another partition. (The only exception occurs when processors fail and are logically replaced for the nonce by more distant processors from another partition.) The design also allows the Data Network to guarantee each processing node at least 5 Mbytes/sec of I/O bandwidth, no matter where it is in the network. However the nodes are divided into partitions, there is always enough Data Network to serve each partition and enough left over to guarantee the stated I/O rate.

When a CM-5 system is first powered up, reset, and bootstrapped, the networks form a single partition that spans the entire system. The operating system then creates a temporary partition for initializing the nodes. It also initializes the I/O devices. After the startup procedures have been completed, administration software establishes one or more operating partitions.

Copyright © 1993 Thinking Machines Corporation

Within each partition, the Network Interfaces are assigned virtual network addresses starting at zero. User programs use virtual network addresses; they are translated by hardware into physical network addresses wherever necessary, in exactly the same way that a memory management unit translates virtual memory addresses to physical memory addresses. Therefore, a user program need not concern itself with the physical network addresses of the partition being used to execute it.

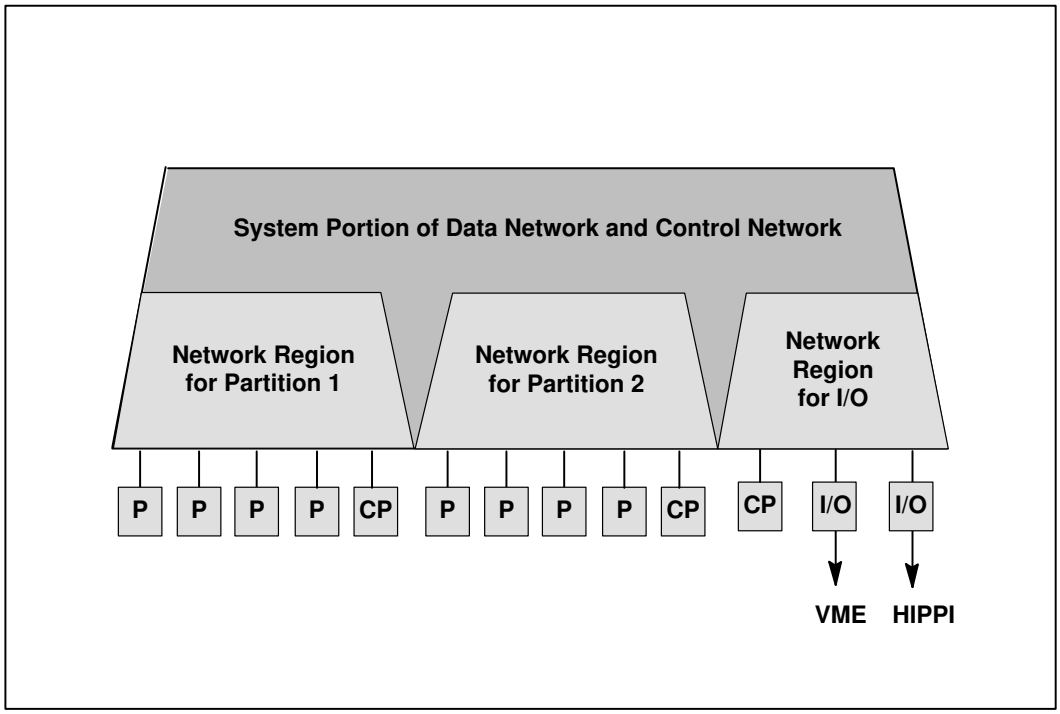

Figure 34. Network support for multiple partitions.

The processing nodes of a CM-5 system can be configured into two or more partitions. Each partition is assigned to a partition manager, a control processor that bears the responsibility for managing the processes executing in that partition. The operating system configures the Control Network and Data Network to match the partition structure. Each partition has a dedicated portion of each network sufficient to provide that partition with the guaranteed minimum network bandwidth of 5 Mbytes/sec for the Control Network and 5 Mbytes/sec per node for the Data Network, regardless of destination. No matter how the partitions are configured, there is always additional network capacity for carrying data between partitions and I/O devices or from one partition to another. Therefore, system-wide data traffic does not interfere with or impede traffic that stays within a partition.

Copyright © 1993 Thinking Machines Corporation

The translation of virtual network addresses includes protection checking that prevents a user process from sending messages to destinations outside its partition. The supervisor can send messages from one partition to another; the mechanism is identical except that it is not subject to the same protection checks because for this purpose the supervisor uses absolute physical network addresses.

I/O is coordinated by the operating system. User processes may transfer data to and from I/O devices, or to and from other user processes (through the facility of UNIX named pipes). In both cases, the operating system breaks up the data into messages and sends the messages through the Data Network. If the two user processes happen to be in the same partition, the message traffic is confined to that partition, not because of protection (the supervisor is responsible for sending the messages in this case) but simply as a consequence of the structure of the Data Network.

19.4 Network Implementation

The topology of the CM-5 Data Network is a *fat tree*, so called because some branches are "fatter" (of higher bandwidth) than others. See Figure 35. This kind of tree is actually more like a biological tree than the computer scientist's usual notion of a tree. A biological tree has skinny twigs, but the limbs are merely slender, the branches are stout, and the trunk is truly fat. Even though there are a thousand twigs and only one trunk, the trunk still has the bandwidth to carry sap for all the twigs because it is fat. (Maybe a computer scientist's tree really ought to be called a "skinny tree" — but of course it's too late now to change the terminology. Computer scientists are also in the habit of drawing trees upside down, with the root at the top and the leaves at the bottom, and we follow this unnatural convention in drawing fat trees here.)

The fat tree structure has a number of distinct advantages over two other topologies, the hypercube and the 2D mesh, used in many other parallel computer systems (including the Connection Machine models CM-1 and CM-2). Like the mesh and hypercube, it can be divided into smaller pieces of the same topology: a mesh can be carved up into smaller meshes; the two halves of a hypercube are themselves hypercubes; and the subtrees of a fat tree are themselves fat trees. This allows the processors to be partitioned in a way that naturally partitions the network, so that each group of processors has its own dedicated portion of the network. Traffic among the processors in one partition does not compete for bandwidth with traffic within another partition.

Copyright © 1993 Thinking Machines Corporation

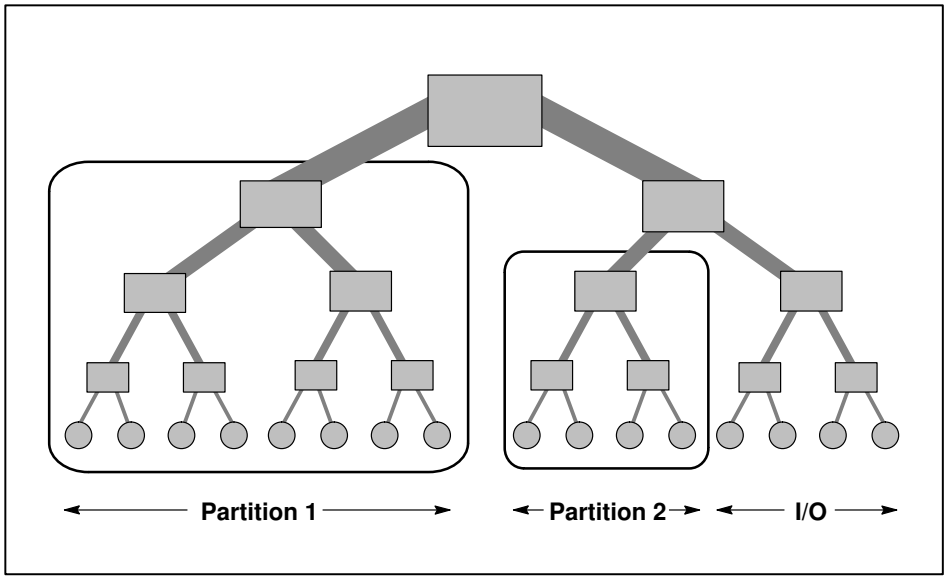

Figure 35. CM-5 Data Network.

The fat tree, however, has an additional property not shared by the other two: traffic between two partitions does not interfere with traffic internal to a third partition. Consider I/O traffic, for example, between partition 1 and the I/O devices in Figure 35. The I/O messages will travel through the upper part of the fat tree, passing *over* partition 2 rather than *through* it, as might happen in a mesh or hypercube topology.

Figure 35 shows only an abstract binary fat tree. The CM-5 Data Network is actually a 4-ary fat tree, where each node has four children. The size of a CM-5 Data Network is often described by its *height*, which is the base-4 logarithm of the number of network addresses spanned. (Put another way, the height of the network equals one-half the number of bits in a processor address.) A CM-5 Scale 3 system, for example, contains a height-3 fat tree, which spans $4^3=64$ network addresses, enough for 32 processing nodes plus control processors and I/O.

Each internal node of the fat tree is implemented as a set of Data Network switches, each a separate VLSI chip. The number of switches per node depends on where it is in the tree; the closer to the root, the fewer nodes and the more switches per node. Each switch has four children and either two or four parents. See Figure 36, which illustrates a fat tree with 16 leaves. Each leaf represents one Network Interface. The level-1 nodes have two switches each; the level-2 nodes

Copyright © 1993 Thinking Machines Corporation

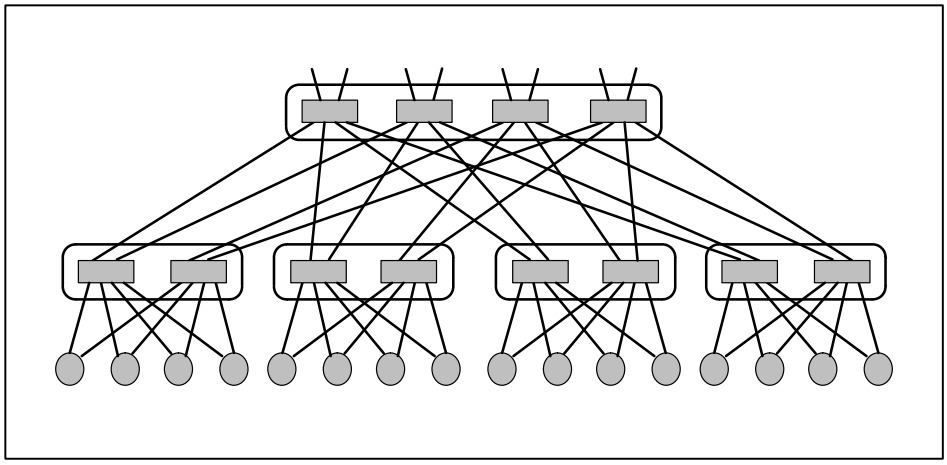

Figure 36. CM-5 fat tree.

have four switches each. As shown in the figure, each level-1 or level-2 switch has two parents. Switches at higher levels have four parents each; at every level above level 2, there are four links going up for every four links coming in from below, thus maintaining constant bandwidth per Network Interface no matter how large the network grows.

The routing algorithm for the Data Network is very simple. The Network Interface compares the physical destination address to its own and determines how far up the tree the message must travel. The message can then take *any* path up the tree. This allows the switches to perform load balancing on the fly. Once the message has reached the necessary height in the tree, it must then follow a particular path down — but the path has the same description no matter what upward path was taken. For example, if a message that has reached height 2 is destined for the tenth processor in Figure 36, then no matter which of the four height-2 switches it has reached, it must travel to the third child (which might be either of two height-1 switches) and then to that child's second child.

Figure 37 is a simplification of Figure 36, showing only the links connecting the switches (and the 16 leaves). Figure 38 then shows how 64 nodes are connected by taking four copies of the 16-node network and adding switches at height 3. Then 256 nodes may be connected by using switches at height 4 to connect four copies of the 64-node network, and so on; see Figure 39, which shows a few height-4 switches and a few wires going up toward height 5.

Copyright © 1993 Thinking Machines Corporation

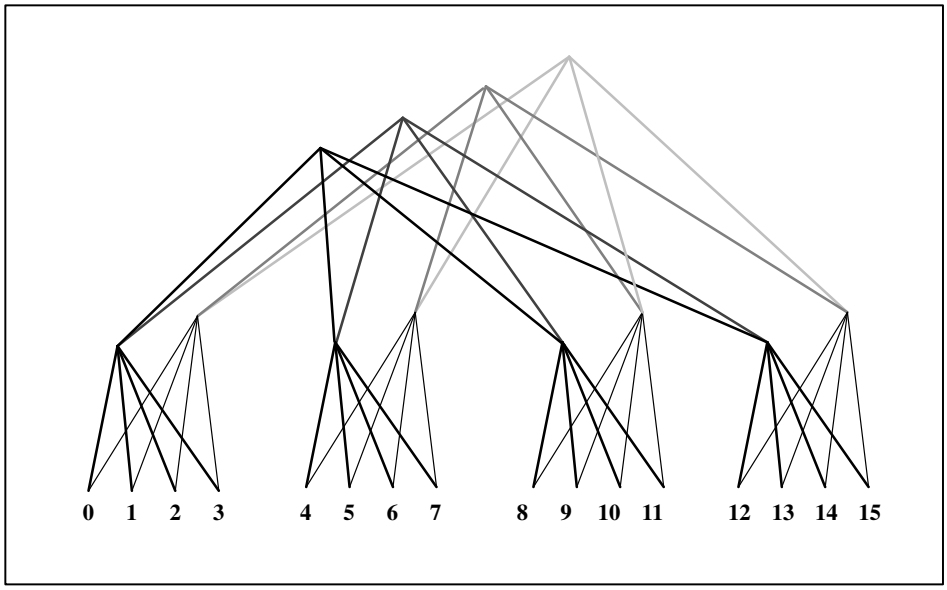

Figure 37. Data Network with 16 nodes.

The CM-5 Data Network has one other good property in addition to scalability, partitionability, and non-interference: it can continue to operate even when one or more switches have failed. See Figure 40. If a link breaks or a switch fails,

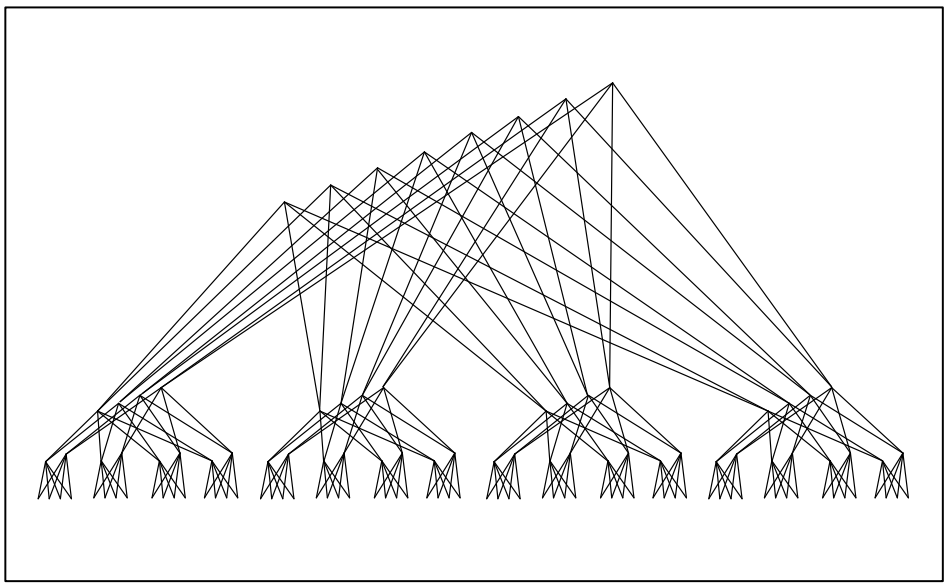

Figure 38. Data Network with 64 nodes.

November 1993
Copyright © 1993 Thinking Machines Corporation

once the problem has been diagnosed then the Diagnostic Network can configure neighboring Data Network chips to ignore the failed components. The switch indicated by the arrow will then have one parent instead of two, so traffic through that switch will see only half the usual bandwidth. However, load balancing at lower levels will react by sending proportionately less traffic through that switch, so that messages soon share seven upgoing wires equally where before they had eight. The result is that the fat tree continues to function with slightly reduced bandwidth.

A mesh with a switch missing is a mesh with a hole in it. In principle traffic could be routed around the missing switch, but the usual routing algorithms for a complete mesh are insufficient; moreover, oblivious routing algorithms, which use

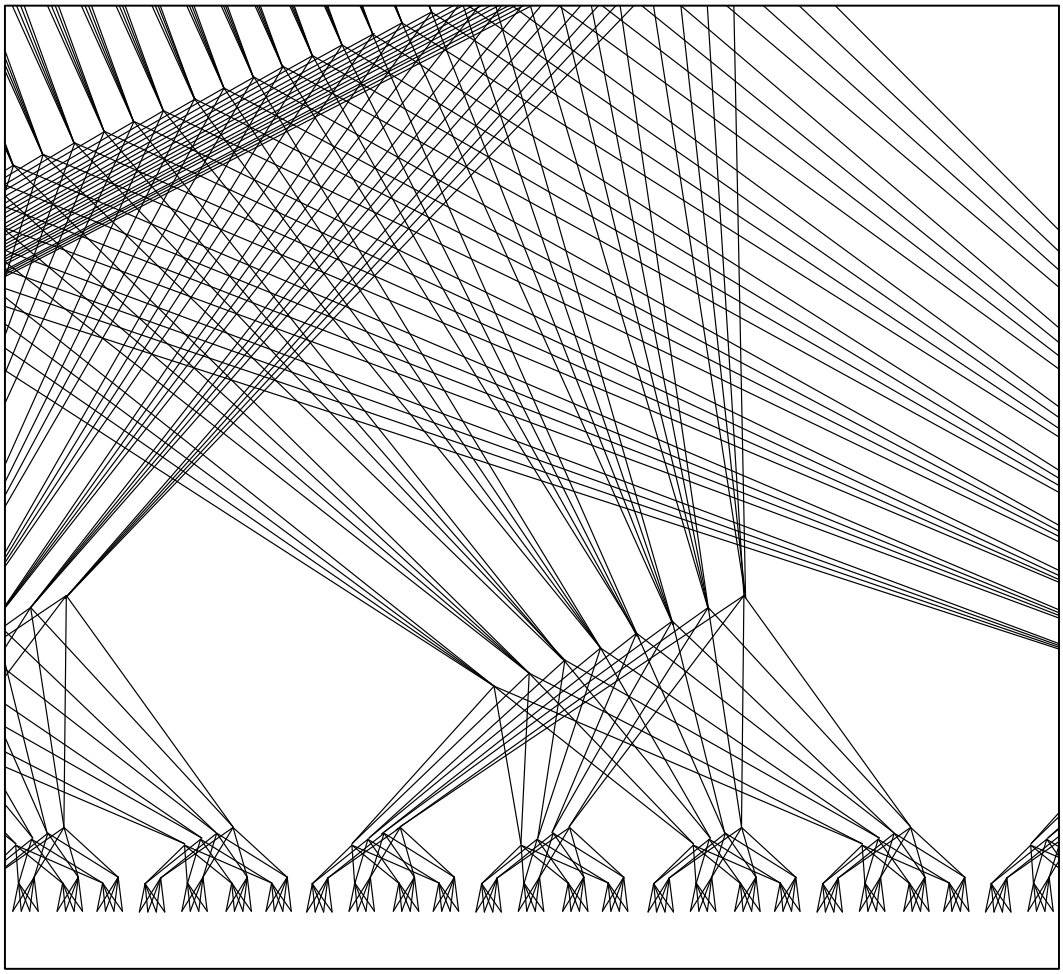

Figure 39. Data Network with 256 nodes.

Copyright © 1993 Thinking Machines Corporation

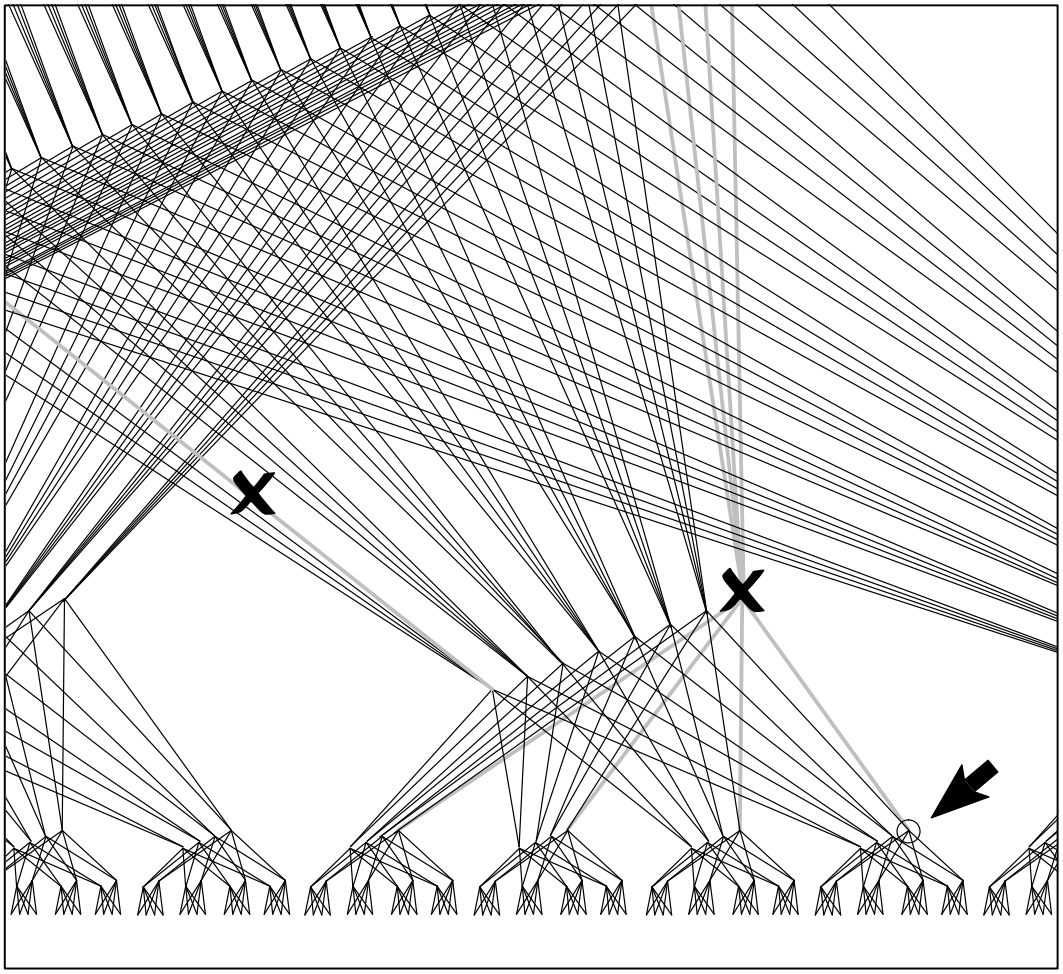

Figure 40. Data Network redundancy.

fixed routing paths rather than reacting on the fly to dynamic network load or configuration, simply haven't a chance. Similarly, a hypercube with a broken switch is a hypercube with a corner missing, and the usual hypercube routing algorithms break down. But a fat tree with missing switches is still a fat tree — it just happens to be a little skinnier in some places — so the standard message routing strategy continues to work.

The Control Network may be thought of as a binary "skinny tree," at least as far as a single user program is concerned. (It is actually a little bit fat, which provides the switching capability necessary to allow any control processor to manage any partition and provides the same ability to function in the face of hardware failures as that of the Data Network.) Each switch in the tree has a

Copyright © 1993 Thinking Machines Corporation

controller, connections to two children and one or two parents, and a small integer arithmetic processor.

Every Control Network operation sends information up the tree to the root (remember, computer scientists draw trees upside down) and then back down from the root to the leaves. There are separate links going up and down, so the Control Network can continuously pipeline information up and down the tree.

To broadcast a message, the Control Network conveys the data from a single Network Interface up the tree to the root; as the data travels back down the tree, it is copied to both children at every switch.

To reduce a set of numbers, each switch waits until both its children have provided inputs; the switch then sends the combined result to its parent. When the root has computed the final combined result, it is then broadcast back to the leaves. Parallel prefix operations are similar but more complicated, with different results traveling back down the tree to different leaves.

The Diagnostic Network is also tree-shaped, with the ability to broadcast diagnostic commands throughout the system and to combine the results of diagnostic tests. Switching at each level of the tree allows selection of arbitrary subsets of chips for diagnosis in parallel. The chips that implement the Diagnostic Network are themselves subject to diagnostic commands; chips lower in the tree may be tested by chips higher in the tree.

19.5 Resource Allocation and Management

CM-5 administration uses standard UNIX mechanisms to control the usage of various resources (disk usage, CPU usage, memory usage, and so on). These are enhanced for the CM-5 when necessary: for instance, stack and heap management can be set for the nodes in a partition as well as for the control processors.

Similarly, standard UNIX procedures govern the mounting and maintenance of file systems.

Copyright © 1993 Thinking Machines Corporation

19.6 Accounting, Monitoring, and Error Reporting

Standard UNIX kernel and device drivers collect information on system activity. Accounting information is collected by ordinary UNIX tools, including NQS, and is logged to a central facility on the System Console.

Errors occurring during normal operation of the CM system are detected by the operating system, collected and distributed by the Control Network.

Hard error signals are collected from the Data Network and from every Network Interface. These signals are combined and distributed according to the current partitioning. Errors detected within a partition are signaled to every Network Interface in that partition, and are reported if appropriate to the user process running at the time of the error. Errors detected in portions of the network outside any partition may be optionally signaled into any designated partition.

The operating system notifies the system administrator of errors by sending a message to the System Console processor. It also logs error information in a central system error log, from which it is available both to the administrator and to diagnostic utilities. System failures and transient hardware errors are also logged to central logging facilities on the System Console.

19.7 Physical Monitoring Systems

The CM-5 system includes extensive power and temperature monitoring systems, designed for early detection of problems that might cause physical damage to the system. The monitoring system reports electronic danger signals, such as detection of an overheating cabinet, to the System Console.

19.8 Fault Detection and Recovery

The CM-5 system is designed to provide high system availability. An important aspect of this design is rapid diagnosis and smooth degradation in the face of component failures. An integrated part of the administration system, the CM-5 diagnostic system is notable for its completeness, its speed, and the high degree of fault isolation it provides. If a failure should occur in a running partition, the administrator can interrogate all items in parallel, isolate the failing item, repartition around the failure, and have the partition up and running again quickly.

November 1993
Copyright © 1993 Thinking Machines Corporation

In addition, the CM-5 provides hardware and software support for checkpointing, either at specified time intervals or by explicit program request. The goal is to allow user applications to be restarted with full system capabilities, even in the presence of failed components.

The Diagnostic Network

The Diagnostic Network, which can probe and control the rest of the system, handles diagnostics. This network is designed to be simple and reliable. It is not particularly fast compared to the Control Network or Data Network, but testing and diagnostic procedures are nevertheless speedy because the Diagnostics Network can operate on all parts of the system in parallel.

The Diagnostic Library

The CM-5 diagnostic library includes a wide variety of tests. Particularly noteworthy among these are the JTAG diagnostics. Based on the IEEE Standard 1149.1, these scan-based vectors both test chips with a very high level of fault coverage and provide connectivity tests between chips (known as boundary scan checking). JTAG diagnostics exist for all CM-5 components; they provide extremely precise isolation of faults. This precision, in turn, allows rapid identification and replacement of failed components, and provides the data necessary for the administration database to exclude failed components when configuring partitions.

Diagnostics are run by partition: thus, one partition can be running diagnostics while others are running user programs. Within the partition, the administrator can choose to run diagnostics on

- the entire partition
- a single subsystem, such as the Control Network
- a single type of component, such as the nodes

Parallel processing provides speed. In the last example, all nodes are tested in parallel and report their status in parallel. In the first example, diagnostic tests on all components of the partition are run in parallel. The Diagnostics Network can address test vectors to the entire system or to any subsystem, such as a backplane, that is believed to be broken. The status of multiple chips or boards

Copyright © 1993 Thinking Machines Corporation

of the same type is read out in parallel, and components whose values differ from an expected value are quickly isolated.

Diagnostics and Components

All CM-5 components are designed to be testable when in place in the system. Nearly all data paths are protected by parity or full CRC. All dynamic memory is protected by full ECC that corrects single-bit errors and detects double-bit errors and DRAM chip failures. Transfers through the Control Network and Data Network are checked by hardware, not merely end-to-end but on every link, so that network component failures can be located precisely.

Failed components can be logically and electrically isolated from the rest of the system under control of the Diagnostics Network. Surrounding components are instructed to ignore any and all signals from failed components. The failed section of the system can then independently execute diagnostic tests or be powered down for repair or replacement, while the rest of the system continues normal operation.

All major CM-5 system components use either redundant or spare component schemes. If a processing node fails, its local group of nodes is taken out of service and can be logically replaced by any other such group from anywhere in the system. All control processors are logically interchangeable; any control processor can manage any partition.

If a Control Network component fails, the consequences depend on the location of the failure within the network. It may be necessary to give up the use of 1/64 of the Network Interfaces in that partition and whatever they are connected to. In this case, spare processors may be logically mapped in to replace them. In other cases, the failure implies the loss of one partition. For example, if a CM-5 system supports up to 8 different partitions, then a Control Network failure might reduce the maximum number of partitions to 7— but the processing resources in the failed partition could be reallocated to other partitions.

If a Data Network component fails, the consequences similarly depend on the location of the failure. It may be necessary to give up the use of 1/64 of the Network Interfaces in that partition and whatever they are connected to. In other cases, no Network Interface need be abandoned; the total global bandwidth of the Data Network is diminished, but never by more than about 6 percent for each failure.

November 1993
Copyright © 1993 Thinking Machines Corporation

I/O devices are also designed to tolerate failures; disk arrays, for example, are designed to tolerate the failure of one or more disk units without loss of data. See the descriptions of individual I/O devices for details.

Copyright © 1993 Thinking Machines Corporation